

A Unified Model for Differential Expression Analysis of RNA-seq Data via L1-Penalized Linear Regression

Kefei Liu, Jieping Ye, *Senior Member, IEEE*, Yang Yang, Li Shen, *Member, IEEE*, and Hui Jiang

Abstract—The RNA-sequencing (RNA-seq) is becoming increasingly popular for quantifying gene expression levels. Since the RNA-seq measurements are relative in nature, between-sample normalization of counts is an essential step in differential expression (DE) analysis. The normalization of existing DE detection algorithms is ad hoc and performed once for all prior to DE detection, which may be suboptimal since ideally normalization should be based on non-DE genes only and thus coupled with DE detection. We propose a unified statistical model for joint normalization and DE detection of log-transformed RNA-seq data. Sample-specific normalization factors are modeled as unknown parameters in the gene-wise linear models and jointly estimated with the regression coefficients. By imposing sparsity-inducing L1 penalty (or mixed L1/L2-norm for multiple treatment conditions) on the regression coefficients, we formulate the problem as a penalized least-squares regression problem and apply the augmented lagrangian method to solve it. Simulation studies show that the proposed model and algorithms outperform existing methods in terms of detection power and false-positive rate when more than half of the genes are differentially expressed and/or when the up- and down-regulated genes among DE genes are unbalanced in amount.

Index Terms—RNA-seq, normalization, differential analysis, augmented Lagrangian method, L1-penalized regression

1 INTRODUCTION

Ultra high-throughput sequencing of transcriptomes (RNA-seq) is a widely used method for quantifying gene expression levels due to its low cost, high accuracy and wide dynamic range for detection [1]. As of today, modern ultra high-throughput sequencing platforms can generate hundreds of millions of sequencing reads from each biological sample in a single day. RNA-seq also facilitates the detection of novel transcripts [2] and the quantification of transcripts on isoform level [3], [4]. For these reasons, RNA-seq has become the method of choice for assaying transcriptomes [5].

One major limitation of RNA-seq is that it only provides relative measurements of transcript abundances due to difference in library size (i.e., sequencing depth) between samples [6]. Normalization of RNA-seq read counts is required in gene differential expression analysis to correct for such variation between samples. A popular form of between-sample normalization is achieved by scaling raw read counts in each sample by a sample-specific factor related to library size [6], [7]. This include CPM/RPM (counts/reads per million) [8], quantile normalization [9], [10], upper-quartile normalization [11], trimmed mean of M

values [8] and DESeq normalization [12]. Also, commonly-used gene expression measures, e.g., TPM (transcript per million) [13], and RPKM/FPKM (reads/fragments per kilobase of exon per million mapped reads) [1], [2], also correct for difference in gene length within a sample [14] (the so-called within-sample normalization). In particular, the CPM/RPM (counts/reads per million) [8], TPM (transcript per million) [13], and RPKM/FPKM (reads/fragments per kilobase of exon per million mapped reads) [1], [2] for the i -th gene from the j -th sample are respectively defined as

$$\begin{aligned} \text{cpm}_{ij} &= 10^6 \frac{c_{ij}}{N_j} \\ \text{fpkm}_{ij} &= 10^9 \frac{c_{ij}}{l_i \cdot N_j} \\ \text{tpm}_{ij} &= 10^6 \frac{c_{ij}/l_i}{\sum_i c_{ij}/l_i} \end{aligned} \quad (1)$$

where c_{ij} is the observed read count for gene i from the j -th sample, $N_j = \sum_i c_{ij}$ is the sequencing depth in the j -th sample, and l_i be the length of gene i . In this work we focus on between-sample normalization.

In traditional count-based RNA-seq analysis methods, the read counts for each gene are assumed to follow a Poisson [15] or negative binomial (NB) distribution. One issue with the count-based RNA-seq analysis methods is that their procedures are complicated and contain many ad hoc heuristics. Moreover, the Poisson or NB distributions of counts are mathematically less tractable than the normal distribution [16], [17]. This makes count-based methods difficult to generalize to new data. Moreover, commonly-used statistical methods for microarray data analysis, e.g., quality weighting of RNA samples, addition of random noise to generate technical replicates, and gene set test [16] have been designed for normally distributed data and it is

- Kefei Liu and Li Shen are with the Department of Radiology and Imaging Sciences, Indiana University School of Medicine, Indianapolis, IN 46202. E-mail: {kefliu,shenli}@iu.edu
- Jieping Ye is with the Department of Computational Medicine and Bioinformatics, University of Michigan, MI 48109. E-mail: jpye@umich.edu
- Yang Yang is with the School of Computer Science and Engineering, Beihang University, Beijing 100191, China. E-mail: yangyangfuture@buaa.edu.cn
- Hui Jiang is with the Department of Biostatistics, University of Michigan, MI 48109. E-mail: jianghui@umich.edu

unclear whether we can adapt them to count data. Also the presence of outliers is an issue that is not well addressed (addressed in a very ad hoc manner) by existing methods. To handle that, the authors of [16] take the logarithm of the raw count of reads and apply normal distribution-based statistical methods to analyze them. Note that by logarithmic transformation, the dynamic range of the RNA-seq counts is compressed such that the outlier counts are largely transformed into “normal” data. As a result, sophisticated way to detect and discard outliers [18], [19], [20] is not required.

In this paper, like in [16], [17] we work with log-transformed gene expression values and propose a unified statistical model for differential gene expression. Different from [16], [17], we model sample-specific scaling factors for between-sample normalization as unknown parameters and incorporate them into the gene-wise linear models. By imposing the sparsity-inducing penalty (ℓ_1 -norm for single treatment factor and mixed ℓ_1/ℓ_2 -norm for multiple treatment factors) on the regression coefficients and carefully choosing the penalty parameter, the model is able to achieve joint accurate detection of DE genes and between-sample normalization. To fit the model, we first eliminate sample-specific parameters using optimization argumentation to formulate the problem as a penalized linear regression problem, and then solve it with the alternating direction method of multipliers algorithm (ADMM), which is known for its fast convergence to modest accuracy [21]. Regarding the choice of penalty parameter, we theoretically derive the smallest penalty parameter α_{\max} that leads to all-zero solution, and thereby find a proper penalty parameter within $[0, \alpha_{\max}]$. Simulation studies show that the proposed methods perform better in terms of detection power and false-positive rate than existing methods when more than 50% of the genes are differentially expressed and/or the fold change distribution is asymmetric¹. Moreover, it is robust against deviations of the distribution of RNA-seq count data and suffers almost no performance degradation even when the data is generated according to the probabilistic assumptions of the previous methods.

Note that our work is preceded by [22] which address the differential expression problem in a similar way. The difference is that the model of [22] considers only categorical or qualitative predictor/explanatory variables (treatment conditions). For example, label “0” is assigned to samples from the control group and label “1” to samples from the treatment group. While in our model, the predictor/explanatory variables can take arbitrary numeric values, and is thus a generalization of [22] from discrete to continuous predictor-variable model case. Note that the algorithm in [22] does not apply to the current numeric variable model at hand, because (i) applicability: it requires that multiple samples are present in each group but in the continuous-predictor model the concept of “group” no longer exists, or more precisely, each group is formed by only one sample; (ii) algorithmic complexity: it requires an p -dimensional exhaustive search, where p is the number of treatment conditions. When $p > 1$ (see Section 4), the algorithm is computationally very

expensive.

The remainder of the paper is organized as follows. In Section 2, we formulate the problem in the context of a single treatment factor. In Section 3, we formulate the problem as a penalized simple regression problem and derive efficient ADMM algorithm to solve it, together with the estimation of noise variance and penalty parameter. In Section 4, we extend the simple regression model to multiple linear regression model. Comparison with existing methods is presented in Section 5, followed by discussions in Section 6.

2 DATA MODEL AND PROBLEM FORMULATION

Throughout the paper, the subscript is used to index the vectors for rows of a matrix. For example, the i -th row vector of a matrix A is denoted as a_i .

2.1 Data model

Suppose there are a total of m genes measured in n samples. Let y_{ij} , $i = 1, 2, \dots, m$, $j = 1, 2, \dots, n$, be the log-transformed gene expression measurements (a small positive number is usually added before taking logarithm) for the i -th gene from the j -th sample. The following statistical model is assumed

$$y_{ij} \sim \mathcal{N}(\beta_{0i} + \beta_i x_j + d_j, \sigma_i^2) \quad (2)$$

where β_{0i} is the y -intercept for gene i , x_j , $j = 1, 2, \dots, n$, is the predictor variable that represents the treatment condition (e.g., drug dosage) for sample j , β_i is the slope or regression coefficient representing log-fold-change of expression levels of gene i with unit change of x_j , d_j is the scaling factor (e.g., log(sequencing depth) or log(library size)) for sample j for between-sample normalization [6], and σ_i is the standard deviation of log-transformed expression levels of gene i .

In (2), we consider a single treatment condition. Extension to models with multiple treatment conditions will be discussed in Section 4.

Our main interest is to detect differentially expressed (DE) genes, i.e., whether β_i is equal to zero. If $\beta_i \neq 0$ gene i is differentially expressed across the n samples; otherwise it is not.

Remark 2.1. Since β_{0i} and d_j in (2) respectively model gene-specific factor (e.g., gene length) and sample-specific factor, model (2) able to work with any log-transformed gene expression measures in the form of

$$y_{ij} = \log \frac{c_{ij}}{l_i \cdot q_j}, \quad (3)$$

where c_{ij} is the raw counts, l_i is the length of gene i and q_j is the normalization factor of the j -th sample, since l_i and q_j can be absorbed into β_{0i} and d_j , respectively. Note that gene expression measures of form $c_{ij}/(l_i \cdot q_j)$ include the raw counts (with $l_i = q_j = 1$), measures based on between-sample normalization only ($l_i = 1$) [6], and FPKM and TPM which are shown in (1) and involve both between- and within-sample normalization.

1. Here by “asymmetric” we mean the up- and down-regulated DE genes are unbalanced in number [(or amount)].

2.2 Penalized likelihood

Since the gene expression measurements y_{ij} are independent across the genes and samples, the likelihood is given by

$$\Pr(\mathbf{y}|\beta_0, \beta, \mathbf{d}) = \prod_{i=1}^m \prod_{j=1}^n \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left\{-\frac{(y_{ij} - \beta_{0i} - \beta_i x_j - d_j)^2}{2\sigma_i^2}\right\}, \quad (4)$$

where

$$\beta = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_m \end{pmatrix}.$$

Assume that $\{\sigma_i^2\}_{i=1}^m$ are known, maximization of (4) is equivalent to minimizing the negative log-likelihood:

$$l(\beta_0, \beta, \mathbf{d}; \mathbf{y}) = \sum_{i=1}^m \sum_{j=1}^n \frac{1}{2\sigma_i^2} (y_{ij} - \beta_{0i} - \beta_i x_j - d_j)^2, \quad (5)$$

where we have ignored the irrelevant constant.

In practice, we solve for $\{\sigma_i^2\}_{i=1}^m$ using an ad hoc approach, which will be described in Section 3.4.

We introduce a ℓ_1 -penalty on the β_i 's:

$$p(\beta) = \alpha \|\beta\|_1 := \alpha \sum_{i=1}^m |\beta_i|. \quad (6)$$

It is well known that the ℓ_1 -norm penalty favors sparse solutions (forces some coefficients to be exactly zero) [23]. This is reasonable since in practice many genes are not differentially expressed.

The objective function to be minimized is

$$f(\beta_0, \beta, \mathbf{d}) = \sum_{i=1}^m \sum_{j=1}^n \frac{1}{2\sigma_i^2} (y_{ij} - \beta_{0i} - x_j \beta_i - d_j)^2 + \alpha \sum_{i=1}^m |\beta_i|. \quad (7)$$

3 ALGORITHM DEVELOPMENT

3.1 Formulation of (7) as Penalized Simple Linear Regression Model

It can be proved that the optimization problem in (7) is jointly convex in $(\beta_0, \beta, \mathbf{d})$. Therefore, the minimizer of (7) is the stationary point.

The derivative of $f(\beta_0, \beta, \mathbf{d})$ with respect to d_j , $j = 1, 2, \dots, n$, is

$$\frac{\partial f}{\partial d_j} = \sum_{i=1}^m -\frac{1}{\sigma_i^2} (y_{ij} - \beta_{0i} - x_j \beta_i - d_j) = 0. \quad (8)$$

Setting (8) to zero gives

$$d_j = \frac{1}{\sum_{i=1}^m \frac{1}{\sigma_i^2}} \sum_{i=1}^m \frac{1}{\sigma_i^2} (y_{ij} - \beta_{0i} - x_j \beta_i). \quad (9)$$

Model (2) is non-identifiable because we can simply add any constant to all the d_j 's and subtract the same constant from all the β_{0i} 's, while having the same fit. To resolve this issue, we fix $d_1 = 0$. Therefore

$$d_j = d_j - d_1 = \left(\bar{y}_{\cdot j}^{(w)} - \bar{y}_{\cdot 1}^{(w)} \right) - (x_j - x_1) \bar{\beta}^{(w)}, \quad (10)$$

where

$$\bar{y}_{\cdot j}^{(w)} := \frac{1}{\sum_{i=1}^m \frac{1}{\sigma_i^2}} \sum_{i=1}^m \frac{1}{\sigma_i^2} y_{ij}, \text{ for } j = 1, 2, \dots, n, \quad (11)$$

$$\bar{\beta}^{(w)} := \frac{1}{\sum_{i=1}^m \frac{1}{\sigma_i^2}} \sum_{i=1}^m \frac{1}{\sigma_i^2} \beta_i. \quad (12)$$

On the other hand, from

$$\frac{\partial f}{\partial \beta_{0i}} = -\frac{1}{\sigma_i^2} \sum_{j=1}^n (y_{ij} - \beta_{0i} - x_j \beta_i - d_j) = 0, \quad (13)$$

we have

$$\beta_{0i} = \frac{1}{n} \sum_{j=1}^n (y_{ij} - x_j \beta_i - d_j) = \bar{y}_{i \cdot} - \bar{x} \beta_i - \frac{1}{n} \sum_{j=1}^n d_j. \quad (14)$$

where

$$\bar{y}_{i \cdot} := \frac{1}{n} \sum_{j=1}^n y_{ij}, \text{ for } i = 1, 2, \dots, m. \quad (15)$$

$$\bar{x} := \frac{1}{n} \sum_{j=1}^n x_j. \quad (16)$$

From (10) we have

$$\frac{1}{n} \sum_{j=1}^n d_j = \left(\bar{y}^{(w)} - \bar{y}_{\cdot 1}^{(w)} \right) - (\bar{x} - x_1) \bar{\beta}^{(w)}, \quad (17)$$

where

$$\bar{y}^{(w)} := \frac{1}{\sum_{i=1}^m \frac{1}{\sigma_i^2}} \sum_{i=1}^m \frac{1}{\sigma_i^2} \cdot \frac{1}{n} \sum_{j=1}^n y_{ij}. \quad (18)$$

Substituting (17) into (14) yields

$$\beta_{0i} = \bar{y}_{i \cdot} + \bar{y}_{\cdot 1}^{(w)} - \bar{y}^{(w)} + (\bar{x} - x_1) \bar{\beta}^{(w)} - \bar{x} \beta_i. \quad (19)$$

Without loss of generality, we make the following assumptions:

Assumption 3.1.

$$\sum_{j=1}^n x_j = n\bar{x} = 0, \quad \sum_{j=1}^n x_j^2 = 1. \quad (20)$$

This assumption is reasonable since in the model (2) the center and scaling factor of x_j 's can be absorbed into β_{0i} and β_i , respectively.

Then (19) simplifies to

$$\beta_{0i} = \bar{y}_{i \cdot} + \bar{y}_{\cdot 1}^{(w)} - \bar{y}^{(w)} - x_1 \bar{\beta}^{(w)}. \quad (21)$$

The sum of (10) and (21) yields

$$\beta_{0i} + d_j = \bar{y}_{i \cdot} + \bar{y}_{\cdot j}^{(w)} - \bar{y}^{(w)} - x_j \bar{\beta}^{(w)}. \quad (22)$$

Substituting (22) into (7), the latter simplifies to

$$f(\beta) = \sum_{i=1}^m \frac{1}{2\sigma_i^2} \sum_{j=1}^n \left(\tilde{y}_{ij} - x_j \beta_i + x_j \bar{\beta}^{(w)} \right)^2 + \alpha \sum_{i=1}^m |\beta_i|, \quad (23)$$

where

$$\tilde{y}_{ij} := y_{ij} - \bar{y}_{i \cdot} - \bar{y}_{\cdot j}^{(w)} + \bar{y}^{(w)}. \quad (24)$$

It can be shown by straightforward calculation that $\{\tilde{y}_{ij}\}$ satisfies

$$\sum_{i=1}^m \frac{1}{\sigma_i^2} \tilde{y}_{ij} = 0. \quad (25)$$

$$\sum_{j=1}^n \tilde{y}_{ij} = 0. \quad (26)$$

3.2 Model Fitting by ADMM

We propose to use the alternating direction method of multipliers (ADMM) [21] to solve (23). Although ADMM can be very slow to converge to high accuracy, it is often the case that ADMM converges to modest accuracy very fast (within a few tens of iterations) [21].

To apply the ADMM, the problem (23) is reformulated as

$$f(\beta) = \sum_{i=1}^m \frac{1}{2\sigma_i^2} \sum_{j=1}^n (\tilde{y}_{ij} - x_j \beta_i + x_j \delta_0)^2 + \alpha \sum_{i=1}^m |\beta_i|, \quad (27a)$$

subject to

$$\frac{1}{\sum_{i=1}^m \frac{1}{\sigma_i^2}} \sum_{i=1}^m \frac{1}{\sigma_i^2} \beta_i = \delta_0. \quad (27b)$$

The augmented Lagrangian of (27) is (28) at the bottom of the page.

Step 1: Update $\beta_i, i = 1, 2, \dots, m$:

The derivative of (28) with respect to β_i is

$$\begin{aligned} \frac{\partial L_\rho}{\partial \beta_i} &= \frac{1}{\sigma_i^2} \sum_{j=1}^n -x_j (\tilde{y}_{ij} - x_j \beta_i + x_j \delta_0) + \alpha \partial |\beta_i| \\ &+ \frac{1}{\sum_{\ell=1}^m \frac{1}{\sigma_\ell^2}} \frac{1}{\sigma_i^2} \lambda + \rho \frac{1}{\sum_{\ell=1}^m \frac{1}{\sigma_\ell^2}} \frac{1}{\sigma_i^2} \left(\frac{1}{\sum_{\ell=1}^m \frac{1}{\sigma_\ell^2}} \sum_{\ell=1}^m \frac{1}{\sigma_\ell^2} \beta_\ell - \delta_0 \right), \end{aligned} \quad (29)$$

where $\partial |\beta_i|$ is the subgradient of $|\beta_i|$ with respect to β_i and is defined as

$$\partial |\beta_i| = \begin{cases} 1, & \beta_i > 0 \\ -1, & \beta_i < 0 \\ [-1, 1], & \beta_i = 0 \end{cases}$$

Setting (29) equal to zero gives (30) at the bottom of the page, where T is the soft-thresholding operator:

$$T_{\sigma_i^2 \alpha} [x] := \text{sign}(x) \left(|x| - \sigma_i^2 \alpha \right)_+ = \begin{cases} x - \sigma_i^2 \alpha, & x > \sigma_i^2 \alpha \\ x + \sigma_i^2 \alpha, & x < -\sigma_i^2 \alpha \\ 0, & -\sigma_i^2 \alpha \leq x \leq \sigma_i^2 \alpha \end{cases}$$

Step 2: Update δ_0 :

The derivative of (28) with respect to δ_0 is

$$\begin{aligned} L_\rho(\beta, \delta_0, \lambda) &= \sum_{i=1}^m \frac{1}{\sigma_i^2} \sum_{j=1}^n x_j (\tilde{y}_{ij} - x_j \beta_i + x_j \delta_0) \\ &- \lambda + \rho \left(\delta_0 - \frac{1}{\sum_{i=1}^m \frac{1}{\sigma_i^2}} \sum_{i=1}^m \frac{1}{\sigma_i^2} \beta_i \right). \end{aligned} \quad (31)$$

Setting (31) equal to zero gives

$$\begin{aligned} \delta_0 &= \frac{1}{\sum_{i=1}^m \frac{1}{\sigma_i^2} + \rho} \left(\lambda - \sum_{i=1}^m \frac{1}{\sigma_i^2} \sum_{j=1}^n x_j \tilde{y}_{ij} \right) + \frac{1}{\sum_{i=1}^m \frac{1}{\sigma_i^2}} \sum_{i=1}^m \frac{1}{\sigma_i^2} \beta_i \\ &= \frac{1}{\sum_{i=1}^m \frac{1}{\sigma_i^2} + \rho} \lambda + \frac{1}{\sum_{i=1}^m \frac{1}{\sigma_i^2}} \sum_{i=1}^m \frac{1}{\sigma_i^2} \beta_i, \end{aligned} \quad (32)$$

where the second equality is due to (25).

Step 3: Update λ :

$$\lambda^{\text{new}} = \lambda^{\text{old}} + \rho \left(\frac{1}{\sum_{i=1}^m \frac{1}{\sigma_i^2}} \sum_{i=1}^m \frac{1}{\sigma_i^2} \beta_i - \delta_0 \right) \quad (33)$$

The model fitting algorithm is described in Algorithm 1.

3.3 Estimation of Penalty Parameter α

The (23) can be expressed in matrix form as

$$f(\beta) = \frac{1}{2} \left\| \Sigma^{1/2} (\tilde{Y} - M\beta x^T) \right\|_F^2 + \alpha \|\beta\|_1, \quad (40)$$

where

$$\Sigma = \text{diag}\{\sigma\}, \quad (41)$$

with

$$\sigma = \left(1/\sigma_1^2 \quad 1/\sigma_2^2 \quad \dots \quad 1/\sigma_m^2 \right)^T, \quad (42)$$

and

$$\begin{aligned} M &= \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \dots & 1 \end{pmatrix} - \frac{1}{\sum_{i=1}^m \frac{1}{\sigma_i^2}} \begin{pmatrix} 1/\sigma_1^2 & 1/\sigma_2^2 & \dots & 1/\sigma_m^2 \\ 1/\sigma_1^2 & 1/\sigma_2^2 & \dots & 1/\sigma_m^2 \\ \vdots & \vdots & \ddots & \vdots \\ 1/\sigma_1^2 & 1/\sigma_2^2 & \dots & 1/\sigma_m^2 \end{pmatrix} \\ &= I_m - \frac{1}{\sum_{i=1}^m \frac{1}{\sigma_i^2}} \mathbf{1}_m \sigma^T. \end{aligned} \quad (43)$$

After expansion, (40) becomes

$$f(\beta) = \frac{1}{2} \left\| \Sigma^{1/2} \tilde{Y} \right\|_F^2 - \beta^T M^T \Sigma \tilde{Y} x + \frac{1}{2} \beta^T M^T \Sigma M \beta + \alpha \|\beta\|_1, \quad (44)$$

where we exploit the assumption $x^T x = 1$.

Since $\frac{1}{2} \beta^T M^T \Sigma M \beta \geq 0$ with equality occurring at $\beta = \mathbf{0}$, it is shown that $\hat{\beta} = \mathbf{0}$ is the minimizer of $f(\beta)$ when

$$\alpha \geq \left\| M^T \Sigma \tilde{Y} x \right\|_\infty := \max_{1 \leq i \leq m} \left| m^i T \Sigma \tilde{Y} x \right|, \quad (45)$$

where m^i denotes the i -th column of M in (43).

Note that

$$M^T \Sigma \tilde{Y} = \left(I_m - \frac{1}{\sum_{i=1}^m \frac{1}{\sigma_i^2}} \sigma \mathbf{1}_m^T \right) \Sigma \tilde{Y} = \Sigma \tilde{Y}, \quad (46)$$

where the last equality holds because $\mathbf{1}_m^T \Sigma \tilde{Y} = \mathbf{0}$ due to (25).

Substituting (46) into (45) yields

$$\alpha_{\max} = \left\| \Sigma \tilde{Y} x \right\|_\infty = \max_j \left| \frac{1}{\sigma_j^2} x^T \tilde{y}_j \right|. \quad (47)$$

Our strategy is to set $\alpha = \epsilon \alpha_{\max}$, where $0 < \epsilon < 1$. Empirically we found that $\epsilon \in [0.001, 0.1]$ works well in a wide range of parameter settings. We set $\epsilon = 0.01$ in Section 5.

Algorithm 1 Alternating direction method of multipliers

Input: Log-transformed gene expression measurements: $\{y_{ij}\}_{i=1}^m\}_{j=1}^n$, predictor variables: $\{x_j\}_{j=1}^n$ and estimated noise variance: $\{\sigma_i^2\}_{i=1}^m$.

1: Transform data. Normalize $\{x_j\}_{j=1}^n$ to zero mean and unit norm:

$$\tilde{x}_j \leftarrow \frac{x_j - \bar{x}}{\sqrt{\sum_{j=1}^n (x_j - \bar{x})^2}}, \text{ with } \bar{x} := \frac{1}{n} \sum_{j=1}^n x_j.$$

Center y_{ij} to zero mean over row index i and column index j : calculate \tilde{y}_{ij} according to (24).

2: *Initialization:* Set $\rho > 0$ to any fixed constant, e.g., $\rho = 1$ [21]; choose the penalty parameter α according to Section 3.3.

3: $k = 0$; randomly initialize $\beta = \beta_i^0$, $i = 1, 2, \dots, m$, $\delta_0 = \delta_0^0$, and $\lambda = \lambda^0$.

4: **repeat**

5: Update β_i , $i = 1, 2, \dots, m$

6: **for** $i = 1, 2, \dots, m$ **do**

$$\beta_i^{k+1} = \frac{\sigma_i^2 \left(\sum_{\ell=1}^m \sigma_\ell^{-2} \right)^2}{\sigma_i^2 \left(\sum_{\ell=1}^m \sigma_\ell^{-2} \right)^2 + \rho} T_{\sigma_i^2 \alpha} \left[\left(\sum_{j=1}^n \tilde{x}_j \tilde{y}_{ij} + \delta_0^k \right) - \frac{\rho}{\sum_{\ell=1}^m \frac{1}{\sigma_\ell^2}} \left(\sum_{\ell=1}^m \frac{1}{\sigma_\ell^2} \beta_\ell^k - \delta_0 + \frac{\lambda^k}{\rho} \right) \right]. \quad (34)$$

7: **end for**

8: Update δ_0 :

$$\delta_0^{k+1} = \frac{1}{\sum_{i=1}^m \frac{1}{\sigma_i^2} + \rho} \lambda^k + \frac{1}{\sum_{i=1}^m \frac{1}{\sigma_i^2}} \sum_{i=1}^m \frac{1}{\sigma_i^2} \beta_i^{k+1}. \quad (35)$$

9: Update λ :

$$\lambda^{k+1} = \lambda^k + \rho \left(\frac{1}{\sum_{i=1}^m \frac{1}{\sigma_i^2}} \sum_{i=1}^m \frac{1}{\sigma_i^2} \beta_i^{k+1} - \delta_0^{k+1} \right) \quad (36)$$

10: $k \leftarrow k + 1$;

11: **until** convergence or maximum number of iterations is reached.

Output: $\beta_i = \beta_i^k$, $i = 1, 2, \dots, m$, $\bar{\beta}^{(w)} = \delta_0^k$, and

$$\beta_{0i} = \bar{y}_{i\cdot} + \bar{y}_{\cdot 1}^{(w)} - \bar{y}^{(w)} - \tilde{x}_1 \bar{\beta}^{(w)}, \quad i = 1, 2, \dots, m \quad (37)$$

$$d_1 = 0, \quad d_j = \left(\bar{y}_{\cdot j}^{(w)} - \bar{y}_{\cdot 1}^{(w)} \right) - (\tilde{x}_j - \tilde{x}_1) \bar{\beta}^{(w)}, \quad j = 2, \dots, n \quad (38)$$

12: Recover the original parameter space:

$$\beta'_{0i} = \beta_{0i} - \frac{\beta_i \bar{x}}{\sqrt{\sum_{j=1}^n (x_j - \bar{x})^2}}, \quad \beta'_i = \frac{\beta_i}{\sqrt{\sum_{j=1}^n (x_j - \bar{x})^2}}, \quad i = 1, 2, \dots, m. \quad (39)$$

3.4 Estimation of $\{\sigma_i^2\}_{i=1}^m$

setting the result to zero, we have

To solve for $\{\sigma_i^2\}_{i=1}^m$, consider the negative log-likelihood function in (4) with $\{\sigma_i^2\}_{i=1}^m$ being unknown parameters as well:

$$\beta_i = \sum_{j=1}^n x_j y_{ij} - \sum_{j=1}^n x_j (\beta_{0i} + d_j). \quad (49)$$

$$l = \sum_{i=1}^m \left[\frac{n}{2} \log(2\pi\sigma_i^2) + \frac{1}{2\sigma_i^2} \sum_{j=1}^n (y_{ij} - \beta_{0i} - x_j \beta_i - d_j)^2 \right]. \quad (48)$$

Substituting (22) into (49) yields

$$\beta_i = \sum_{j=1}^n x_j y_{ij} - \frac{1}{\sum_{i=1}^m \frac{1}{\sigma_i^2}} \sum_{i=1}^m \frac{1}{\sigma_i^2} \sum_{j=1}^n x_j y_{ij} + \bar{\beta}^{(w)}, \quad (50)$$

Taking partial derivatives of $l(\cdot)$ with respect to d_j and β_{0i} and setting the results to zero, we arrive at (10) and (21) respectively. The sum of (10) and (21) gives (22).

Taking partial derivatives of $l(\cdot)$ with respect to β_i and where $\bar{\beta}^{(w)}$ is defined in (12).

Taking partial derivatives of σ_i^2 and setting the result to zero gives

$$\sigma_i^2 = \frac{1}{n} \sum_{j=1}^n (y_{ij} - \beta_{0i} - x_j \beta_i - d_j)^2. \quad (51)$$

Substituting (22) into (51) yields

$$\sigma_i^2 = \frac{1}{n} \sum_{j=1}^n \left(y_{ij} - \bar{y}_{i\cdot} - \bar{y}_{\cdot j}^{(w)} + \bar{y}^{(w)} - x_j \beta_i + x_j \bar{\beta}^{(w)} \right)^2, \quad (52)$$

where $\bar{y}_{i\cdot}$, $\bar{y}_{\cdot j}^{(w)}$ and $\bar{y}^{(w)}$ are defined in (15), (11) and (18), respectively.

Given initial estimates for $\bar{\beta}^{(w)}$ and $\{\sigma_i^2\}_{i=1}^m$, we can alternate equations (50), (52) and (12) iteratively to gradually refine the estimates for β_i and σ_i^2 , as shown in Algorithm 2.

Algorithm 2 Estimation of $\{\sigma_i^2\}_{i=1}^m$

Input: Log-transformed gene expression measurements:

$\{y_{ij}\}_{i=1}^m\}_{j=1}^n$ and predictor variables: $\{x_j\}_{j=1}^n$.

1: Normalize $\{x_j\}_{j=1}^n$ to zero mean and unit norm:

$$x_j \leftarrow \frac{x_j - \bar{x}}{\sqrt{\sum_{j=1}^n (x_j - \bar{x})^2}}, \text{ with } \bar{x} := \frac{1}{n} \sum_{j=1}^n x_j.$$

2: **Initialization:** $\bar{\beta}^{(w)} = 0$, $\sigma_1^2 = \sigma_2^2 = \dots = \sigma_m^2 = 1$.

3: **repeat**

4: **for** $i = 1, 2, \dots, m$ **do**

 Update β_i , $i = 1, 2, \dots, m$ according to (50).

5: **end for**

6: Update $\{\sigma_i^2\}_{i=1}^m$ according to (52).

7: Update $\bar{\beta}^{(w)}$ according to (12).

8: **until** convergence or maximum number of iterations is reached.

Output: $\hat{\sigma}_i^2 = \sigma_i^2$, $i = 1, 2, \dots, m$.

Then we take another weighted average of $\hat{\sigma}_i^2$ and the estimated mean variance across all the genes to obtain a robust estimate for σ_i^2 . That is

$$\hat{\sigma}_i'^2 = (1 - w) \hat{\sigma}_i^2 + w \bar{\hat{\sigma}}^2 \quad (53)$$

where

$$\bar{\hat{\sigma}}^2 = \frac{1}{m} \sum_{i=1}^m \hat{\sigma}_i^2, \quad (54)$$

and the weight w is calculated using the following formula as suggested in [24] which is based on an empirical Bayes approach

$$w = \frac{2(m-1)}{n+1} \left(\frac{1}{m} + \frac{(\bar{\hat{\sigma}}^2)^2}{\sum_{i=1}^m (\hat{\sigma}_i^2 - \bar{\hat{\sigma}}^2)^2} \right). \quad (55)$$

This kind of variance estimation approach is widely used in differential gene expression analysis with small sample sizes [25], [26]. The estimated variances $\hat{\sigma}_i'^2$, $i = 1, 2, \dots, m$, can then be used in Algorithm 1 to solve for $\{\beta_i\}_{i=1}^m$.

Remark 3.1. In the special case of $\sigma_1^2 = \sigma_2^2 = \dots = \sigma_m^2 = \sigma^2$, it no longer requires to estimate σ^2 since the unknown σ^2 in (7) can be absorbed into the penalty parameter α .

4 EXTENSION TO MULTIPLE LINEAR REGRESSION MODEL AND ALGORITHM DEVELOPMENT

In the multiple linear regression model, each response or outcome is modeled by $p > 1$ predictors:

$$y_{ij} \sim \mathcal{N}(\beta_{0i} + \beta_i^T \mathbf{x}_j + d_j, \sigma_i^2) \quad (56)$$

where

$$\beta_i = \begin{pmatrix} \beta_{i1} \\ \beta_{i2} \\ \vdots \\ \beta_{ip} \end{pmatrix} \in \mathbb{R}^{p \times 1} \quad (57)$$

is a vector of regression coefficients representing log-fold-change of expression levels of gene i between treatment conditions, and

$$\mathbf{x}_j = \begin{pmatrix} x_{j1} \\ x_{j2} \\ \vdots \\ x_{jp} \end{pmatrix} \in \mathbb{R}^{p \times 1} \quad (58)$$

is a vector of independent/explanatory variables representing the treatment conditions (drug dosage, blood pressure, age, BMI, etc.) for sample j , and β_{0i} , d_j , and σ_i respectively represent the y -intercept of gene i , scaling factor for sample j and standard deviation of log-transformed expression levels of gene i , as defined in the simple regression model.

Since the sample values are independent across the genes and samples, the likelihood is given by

$$\Pr(\mathbf{y} | \beta_0, \{\beta_i\}, \mathbf{d}) = \prod_{i=1}^m \prod_{j=1}^n \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp \left\{ -\frac{(y_{ij} - \beta_{0i} - \beta_i^T \mathbf{x}_j - d_j)^2}{2\sigma_i^2} \right\}. \quad (59)$$

Assume that $\{\sigma_i^2\}_{i=1}^m$ are known, maximization of (59) leads to minimizing the negative log-likelihood:

$$l(\beta_0, \{\beta_i\}, \mathbf{d}; \mathbf{y}) = \sum_{i=1}^m \sum_{j=1}^n \frac{1}{2\sigma_i^2} (y_{ij} - \beta_{0i} - \beta_i^T \mathbf{x}_j - d_j)^2 \quad (60)$$

The objective function to be minimized is

$$f(\beta_0, \{\beta_i\}, \mathbf{d}) = \sum_{i=1}^m \sum_{j=1}^n \frac{1}{2\sigma_i^2} (y_{ij} - \beta_{0i} - \mathbf{x}_j^T \beta_i - d_j)^2 + \sum_{i=1}^m p(\beta_i). \quad (61)$$

Below we introduce two types of penalty function $p(\beta_i)$.

1) Type I penalty:

$$p(\beta_i) = \alpha |\beta_{ip}|. \quad (62)$$

Gene i is differentially expressed if $\beta_{ip} \neq 0$ and not otherwise. This penalty is for the applications where one covariate is of main interest (e.g., treatment) while we want to adjust for all possible effects of other confounding covariates (e.g., age, gender, etc).

2) Type II penalty:

$$p(\beta_i) = \alpha \|\beta_i\|. \quad (63)$$

Gene i is differentially expressed if $\beta_i \neq 0$ and not otherwise. This penalty is for the applications where all covariates are of interest and we want to identify the genes for which at least one covariate has an effect.

It can be proved that the optimization problem (61) with penalty (62) or (63) is jointly convex in $(\beta_0, \{\beta_i\}, \mathbf{d})$.

Assume that

$$\sum_{j=1}^n \mathbf{x}_j = \mathbf{0}, \quad (64)$$

and set $d_1 = 0$. Using similar argumentation as in Section 3.1 to eliminate β_0 and \mathbf{d} , we simplify (61) to

$$f(\{\beta_i\}) = \sum_{i=1}^m \frac{1}{2\sigma_i^2} \sum_{j=1}^n \left(\tilde{y}_{ij} - \mathbf{x}_j^T \beta_i + \mathbf{x}_j^T \bar{\beta}^{(w)} \right)^2 + \sum_{i=1}^m p(\beta_i), \quad (65)$$

where \tilde{y}_{ij} is the same as that in (24), and

$$\bar{\beta}^{(w)} := \frac{1}{\sum_{i=1}^m \frac{1}{\sigma_i^2}} \sum_{i=1}^m \frac{1}{\sigma_i^2} \beta_i. \quad (66)$$

4.1 Regression with Type I Penalty: Model Fitting by ADMM

To apply the ADMM, we reformulate the Type I penalized regression problem as

$$f(\{\beta_i\}, \delta_0) = \sum_{i=1}^m \frac{1}{2\sigma_i^2} \sum_{j=1}^n \left(\tilde{y}_{ij} - \mathbf{x}_j^T \beta_i + \mathbf{x}_j^T \delta_0 \right)^2 + \alpha \sum_{i=1}^m |\beta_{ip}|, \quad (67a)$$

subject to

$$\frac{1}{\sum_{i=1}^m \frac{1}{\sigma_i^2}} \sum_{i=1}^m \frac{1}{\sigma_i^2} \beta_i = \delta_0. \quad (67b)$$

The augmented Lagrangian of (67) is (65) at the bottom of the page.

Step 1: Update β_i , $i = 1, 2, \dots, m$:

Taking partial derivative of (65) with respect to β_i and setting the result to zero gives

$$\left[\frac{1}{\sigma_i^2} \mathbf{X}^T \mathbf{X} + \frac{\rho}{\sigma_i^4} \left(\sum_{\ell=1}^m \frac{1}{\sigma_\ell^2} \right)^2 \mathbf{I}_p \right] \beta_i + \alpha \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} = \mathbf{v}_i. \quad (69)$$

where

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_n^T \end{pmatrix} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} \in \mathbb{R}^{n \times p}, \quad (70)$$

$\partial |\beta_{ip}|$ is the subgradient of $|\beta_{ip}|$ with respect to β_{ip} , and

$$\mathbf{v}_i = \frac{1}{\sigma_i^2} \left(\sum_{j=1}^n \mathbf{x}_j \tilde{y}_{ij} + \mathbf{X}^T \mathbf{X} \delta_0 \right) - \frac{\rho}{\sigma_i^2} \frac{1}{\sum_{\ell=1}^m \frac{1}{\sigma_\ell^2}} \left(\frac{1}{\sum_{\ell=1}^m \frac{1}{\sigma_\ell^2}} \sum_{\ell \neq i} \frac{1}{\sigma_\ell^2} \beta_\ell - \delta_0 + \frac{\lambda}{\rho} \right), \quad (71)$$

Using the block matrix representation:

$$\frac{1}{\sigma_i^2} \mathbf{X}^T \mathbf{X} + \frac{\rho}{\sigma_i^4} \left(\sum_{\ell=1}^m \frac{1}{\sigma_\ell^2} \right)^2 \mathbf{I}_p = \mathbf{Q} = \begin{pmatrix} \tilde{\mathbf{Q}} & \mathbf{q} \\ \mathbf{q}^T & q_{pp} \end{pmatrix},$$

$$\beta_i = \begin{pmatrix} \beta_i^- \\ \beta_{ip} \end{pmatrix}, \quad \mathbf{v}_i = \begin{pmatrix} \mathbf{v}_i^- \\ v_{ip} \end{pmatrix},$$

where $\tilde{\mathbf{Q}}$ is the submatrix of \mathbf{Q} with last row and last column deleted, from (69) we have

$$\tilde{\mathbf{Q}} \beta_i^- + \mathbf{q} \beta_{ip} = \mathbf{v}_i^- \quad (72)$$

$$\mathbf{q}^T \beta_i^- + q_{pp} \beta_{ip} + \alpha \partial |\beta_{ip}| = v_{ip}. \quad (73)$$

From (72) it follows

$$\beta_i^- = \tilde{\mathbf{Q}}^{-1} (\mathbf{v}_i^- - \mathbf{q} \beta_{ip}). \quad (74)$$

Substituting (74) into (73) yields

$$\beta_{ip} = \frac{1}{q_{pp} - \mathbf{q}^T \tilde{\mathbf{Q}}^{-1} \mathbf{q}} T_\alpha [v_{ip} - \mathbf{q}^T \tilde{\mathbf{Q}}^{-1} \mathbf{v}_i^-]. \quad (75)$$

Step 2: Update δ_0 :

Taking the derivative of (65) with respect to δ_0 and setting the result to zero gives

$$\delta_0 = \left(\sum_{i=1}^m \frac{1}{\sigma_i^2} \mathbf{X}^T \mathbf{X} + \rho \mathbf{I}_p \right)^{-1} \lambda + \frac{1}{\sum_{i=1}^m \frac{1}{\sigma_i^2}} \sum_{i=1}^m \frac{1}{\sigma_i^2} \beta_i, \quad (76)$$

where we have exploited (25).

Step 3: Update λ :

$$\lambda^{\text{new}} = \lambda^{\text{old}} + \rho \left(\frac{1}{\sum_{i=1}^m \frac{1}{\sigma_i^2}} \sum_{i=1}^m \frac{1}{\sigma_i^2} \beta_i - \delta_0 \right). \quad (77)$$

4.2 Regression with Type II Penalty: Model Fitting by ADMM

The Type II penalized regression problem is reformulated as

$$f(\{\beta_i\}, \delta_0) = \sum_{i=1}^m \frac{1}{2\sigma_i^2} \sum_{j=1}^n \left(\tilde{y}_{ij} - \mathbf{x}_j^T \beta_i + \mathbf{x}_j^T \delta_0 \right)^2 + \alpha \sum_{i=1}^m \|\beta_i\|, \quad (78a)$$

subject to

$$\frac{1}{\sum_{i=1}^m \frac{1}{\sigma_i^2}} \sum_{i=1}^m \frac{1}{\sigma_i^2} \beta_i = \delta_0. \quad (78b)$$

The augmented Lagrangian of (78) is (76) at the bottom of the page.

Step 1: Update β_i , $i = 1, 2, \dots, m$:

The relevant terms to compute the derivatives of (76) with respect to β_i is (77) at the bottom of the page, where c is an irrelevant constant which does not depend on β_i , and \mathbf{v}_i is defined in (71).

It can be shown when $\|\mathbf{v}_i\| \leq \alpha$ then $\beta_i = \mathbf{0}$; otherwise denote the eigendecomposition of $\mathbf{X}^T \mathbf{X}$ as $\mathbf{X}^T \mathbf{X} = \mathbf{U} \mathbf{D} \mathbf{U}^T$, we have that minimization of (77) is equivalent to

$$\min_{\beta_i} \frac{1}{2} \|\mathbf{Z} \beta_i - \mathbf{b}_i\|^2 + \alpha \|\beta_i\|. \quad (81a)$$

where

$$\mathbf{Z} = \left[\frac{1}{\sigma_i^2} \mathbf{D} + \frac{\rho}{\sigma_i^4} \frac{1}{\left(\sum_{\ell=1}^m \frac{1}{\sigma_\ell^2} \right)^2} \mathbf{I}_p \right]^{1/2} \mathbf{U}^T, \quad (81b)$$

$$\mathbf{b}_i = \left[\frac{1}{\sigma_i^2} \mathbf{D} + \frac{\rho}{\sigma_i^4} \frac{1}{\left(\sum_{\ell=1}^m \frac{1}{\sigma_\ell^2} \right)^2} \mathbf{I}_p \right]^{-1/2} \mathbf{U}^T \mathbf{v}_i. \quad (81c)$$

As in [27], we use a coordinate descent procedure to optimize (81). For each s , given the estimate of $\{\hat{\beta}_{i\ell}\}_{\ell \neq s}$, β_{is} can be estimated by solving

$$\min_{\beta_{is}} \frac{1}{2} \left\| \mathbf{z}_s \beta_{is} - \mathbf{r}_i^{(s)} \right\|^2 + \alpha \sqrt{\beta_{is}^2 + \sum_{\ell \neq s} \hat{\beta}_{i\ell}^2}, \quad (82)$$

where

$$\mathbf{r}_i^{(s)} = \mathbf{b}_i - \sum_{\ell \neq s} \mathbf{z}_\ell \beta_{i\ell}. \quad (83)$$

We solve (82) via a one-dimensional search. Note that the solution to (82) falls between 0 and $\beta_{is}^o = \mathbf{z}_s^T \mathbf{r}_i^{(s)} / \|\mathbf{z}_s\|^2$, the ordinary least-squares estimate. We can the optimize function in the R package, or `fminbnd` function in MATLAB, which perform one-dimensional search based on golden section search and successive parabolic interpolation.

After updating $\{\beta_i\}_{i=1}^m$, the updates of δ_0 and λ turn out to be the same as that in Section 4.1.

4.3 Estimation of Penalty Parameter α

The (65) can be expressed in matrix form as

$$f(\mathbf{B}) = \frac{1}{2} \left\| \Sigma^{1/2} (\tilde{\mathbf{Y}} - \mathbf{M} \mathbf{B} \mathbf{X}^T) \right\|_F^2 + p(\mathbf{B}), \quad (84)$$

where \mathbf{M} and \mathbf{X} are respectively defined in (43) and (70), and

$$\mathbf{B} = \begin{pmatrix} \beta_1^T \\ \beta_2^T \\ \vdots \\ \beta_m^T \end{pmatrix} = \begin{pmatrix} \beta_{11} & \beta_{12} & \cdots & \beta_{1p} \\ \beta_{21} & \beta_{22} & \cdots & \beta_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{m1} & \beta_{m2} & \cdots & \beta_{mp} \end{pmatrix} \in \mathbb{R}^{m \times p}, \quad (85)$$

and $p(\mathbf{B})$ is the penalty function.

The derivative of $f(\mathbf{B})$ with respect to \mathbf{B} is

$$\frac{\partial f}{\partial \mathbf{B}} = \mathbf{M}^T \Sigma \mathbf{M} \mathbf{B} \mathbf{X}^T \mathbf{X} - \mathbf{M}^T \Sigma \tilde{\mathbf{Y}} \mathbf{X} + \frac{\partial p(\mathbf{B})}{\partial \mathbf{B}}. \quad (86)$$

4.3.1 Type I Penalty

When $p(\mathbf{B}) = \alpha \sum_{i=1}^m |\beta_{ip}|$, its derivative with respect to \mathbf{B} is

$$\frac{\partial p(\mathbf{B})}{\partial \mathbf{B}} = \alpha \begin{pmatrix} 0 & \cdots & 0 & \partial |\beta_{1p}| \\ 0 & \cdots & 0 & \partial |\beta_{2p}| \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & \partial |\beta_{mp}| \end{pmatrix} = \begin{pmatrix} \mathbf{0}_{m \times (p-1)} & \alpha \frac{\partial \|\beta^p\|_1}{\partial \beta^p} \end{pmatrix}. \quad (87)$$

Denote

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}^1 & \cdots & \mathbf{x}^{p-1} & \mathbf{x}^p \end{pmatrix} = \begin{pmatrix} \mathbf{X}_1 & \mathbf{x}^p \end{pmatrix},$$

$$\mathbf{B} = \begin{pmatrix} \beta^1 & \cdots & \beta^{p-1} & \beta^p \end{pmatrix} = \begin{pmatrix} \mathbf{B}_1 & \beta^p \end{pmatrix}.$$

Setting (86) equal to zero gives

$$\mathbf{M}^T \Sigma \mathbf{M} (\mathbf{B}_1 \mathbf{X}_1^T + \beta^p \mathbf{x}^{pT}) \mathbf{X}_1 = \mathbf{M}^T \Sigma \tilde{\mathbf{Y}} \mathbf{X}_1 \quad (88)$$

$$\mathbf{M}^T \Sigma \mathbf{M} (\mathbf{B}_1 \mathbf{X}_1^T + \beta^p \mathbf{x}^{pT}) \mathbf{x}^p + \alpha \frac{\partial \|\beta^p\|_1}{\partial \beta^p} = \mathbf{M}^T \Sigma \tilde{\mathbf{Y}} \mathbf{x}^p. \quad (89)$$

Since $\mathbf{M}^T \Sigma \mathbf{M}$ is rank deficient², the solution to (88) is not unique. We apply the pseudoinverse of $\mathbf{M}^T \Sigma \mathbf{M}$ to obtain the minimum-norm solution to (88):

$$\mathbf{B}_1 = (\mathbf{M}^T \Sigma \mathbf{M})^\dagger (\mathbf{M}^T \Sigma \tilde{\mathbf{Y}} - \mathbf{M}^T \Sigma \mathbf{M} \beta^p \mathbf{x}^{pT}) \mathbf{X}_1 (\mathbf{X}_1^T \mathbf{X}_1)^{-1}. \quad (90)$$

Substituting (90) into (89) yields

$$\begin{aligned} & \mathbf{M}^T \Sigma \mathbf{M} \beta^p \mathbf{x}^{pT} \left[\mathbf{I}_n - \mathbf{X}_1 (\mathbf{X}_1^T \mathbf{X}_1)^{-1} \mathbf{X}_1^T \right] \mathbf{x}^p + \alpha \frac{\partial \|\beta^p\|_1}{\partial \beta^p} \\ &= \mathbf{M}^T \Sigma \tilde{\mathbf{Y}} \left[\mathbf{I}_n - \mathbf{X}_1 (\mathbf{X}_1^T \mathbf{X}_1)^{-1} \mathbf{X}_1^T \right] \mathbf{x}^p. \end{aligned} \quad (91)$$

Note that to arrive at (91), we have exploited the fact that $(\mathbf{M}^T \Sigma \mathbf{M}) (\mathbf{M}^T \Sigma \mathbf{M})^\dagger \mathbf{M}^T \Sigma = \mathbf{M}^T \Sigma$ which is due to that $\mathbf{M}^T \Sigma \mathbf{M} = \mathbf{M}^T \Sigma$ according to the definition of \mathbf{M} in (43) and the definition of the pseudoinverse of a matrix.

Since the coefficient matrix of β^p , i.e., $\mathbf{M}^T \Sigma \mathbf{M} \cdot (\mathbf{x}^{pT} [\mathbf{I}_n - \mathbf{X}_1 (\mathbf{X}_1^T \mathbf{X}_1)^{-1} \mathbf{X}_1^T] \mathbf{x}^p)$ is positive semidefinite, (91) implies that when

$$\begin{aligned} \alpha &\geq \left\| \mathbf{M}^T \Sigma \tilde{\mathbf{Y}} \left[\mathbf{I}_n - \mathbf{X}_1 (\mathbf{X}_1^T \mathbf{X}_1)^{-1} \mathbf{X}_1^T \right] \mathbf{x}^p \right\|_\infty \\ &= \left\| \Sigma \tilde{\mathbf{Y}} \left[\mathbf{I}_n - \mathbf{X}_1 (\mathbf{X}_1^T \mathbf{X}_1)^{-1} \mathbf{X}_1^T \right] \mathbf{x}^p \right\|_\infty \\ &= \max_j \left| \frac{1}{\sigma_i^2} \tilde{\mathbf{y}}_i^T \left[\mathbf{I}_n - \mathbf{X}_1 (\mathbf{X}_1^T \mathbf{X}_1)^{-1} \mathbf{X}_1^T \right] \mathbf{x}^p \right|, \end{aligned} \quad (92)$$

where the next to last equality is due to (46), we obtain zero solution.

4.3.2 Type II Penalty

The derivative of $p(\mathbf{B}) = \alpha \sum_{i=1}^m \|\beta_i\|$ with respect to \mathbf{B} is

$$\frac{\partial p(\mathbf{B})}{\partial \mathbf{B}} = \alpha \begin{pmatrix} \frac{\partial \|\beta_1\|}{\partial \beta_1^T} \\ \frac{\partial \|\beta_2\|}{\partial \beta_2^T} \\ \vdots \\ \frac{\partial \|\beta_m\|}{\partial \beta_m^T} \end{pmatrix}, \quad (93)$$

where $\frac{\partial \|\beta_i\|}{\partial \beta_i} = \frac{\beta_i}{\|\beta_i\|}$ if $\beta_i \neq \mathbf{0}$ and $\left\| \frac{\partial \|\beta_i\|}{\partial \beta_i} \right\| \leq 1$ otherwise [27], [28].

Setting (86) equal to zero yields

$$\mathbf{X}^T \mathbf{X} \mathbf{B}^T \mathbf{M}^T \Sigma \mathbf{m}^i - \mathbf{X}^T \tilde{\mathbf{Y}}^T \Sigma \mathbf{m}^i + \alpha \frac{\partial \|\beta_i\|}{\partial \beta_i} = \mathbf{0}_{p \times 1}, \quad (94)$$

for $i = 1, 2, \dots, m$, where \mathbf{m}^i is the i -th column of \mathbf{M} in (43). The minimizer to $f(\mathbf{B})$ is a zero matrix when

$$\alpha \geq \max_j \left\| \mathbf{X}^T \tilde{\mathbf{Y}}^T \Sigma \mathbf{m}^i \right\|. \quad (95)$$

2. Simple calculation shows that the rank of $\mathbf{M}^T \Sigma \mathbf{M}$ is $m - 1$.

Note that

$$\tilde{\mathbf{Y}}^T \Sigma \mathbf{m}^i = \tilde{\mathbf{Y}}^T \Sigma \left(\mathbf{e}_i - \frac{1}{\sum_{\ell=1}^m \frac{1}{\sigma_\ell^2}} \frac{1}{\sigma_i^2} \mathbf{1}_m \right) = \tilde{\mathbf{Y}}^T \Sigma \mathbf{e}_i = \frac{1}{\sigma_i^2} \tilde{\mathbf{y}}_i, \quad (96)$$

where the next to last equality is due to (25).

Substituting (96) into (95) yields

$$\alpha_{\max} = \max_j \left\| \frac{1}{\sigma_i^2} \mathbf{X}^T \tilde{\mathbf{y}}_i \right\|. \quad (97)$$

4.4 Estimation of $\{\sigma_i^2\}_{i=1}^m$

To solve for $\{\sigma_i^2\}_{i=1}^m$, consider the negative log-likelihood function with $\{\sigma_i^2\}_{i=1}^m$ being unknown parameters as well:

$$l(\beta_0, \{\beta_i\}, \mathbf{d}; \mathbf{y}) = \sum_{i=1}^m \left[\frac{n}{2} \log(2\pi\sigma_i^2) + \frac{1}{2\sigma_i^2} \sum_{j=1}^n (y_{ij} - \beta_{0i} - \mathbf{x}_j^T \beta_i - d_j) \right] \quad (98)$$

Taking partial derivatives of $l(\cdot)$ with respect to d_j and β_{0i} and setting the result to zero, we arrive at

$$d_j = d_j - d_1 = (\bar{y}_{\cdot j}^{(w)} - \bar{y}_{\cdot 1}^{(w)}) - (\mathbf{x}_j - \mathbf{x}_1)^T \bar{\beta}^{(w)}, \quad (99)$$

$$\beta_{0i} = \bar{y}_{i\cdot} - \frac{1}{n} \sum_{j=1}^n \mathbf{x}_j^T \beta_i - \frac{1}{n} \sum_{j=1}^n d_j = \bar{y}_{i\cdot} + \bar{y}_{\cdot 1}^{(w)} - \bar{y}^{(w)} - \mathbf{x}_1^T \bar{\beta}^{(w)}. \quad (100)$$

where to derive the second equality we have exploited assumption (64).

The sum of (99) and (100) gives

$$\beta_{0i} + d_j = \bar{y}_{i\cdot} + \bar{y}_{\cdot j}^{(w)} - \bar{y}^{(w)} - \mathbf{x}_j^T \bar{\beta}^{(w)}. \quad (101)$$

Taking partial derivatives of $l(\cdot)$ with respect to β_i and setting the result to zero, we have

$$\beta_i = \sum_{j=1}^n \mathbf{x}_j y_{ij} - \sum_{j=1}^n \mathbf{x}_j (\beta_{0i} + d_j). \quad (102)$$

Substituting (101) into (102) yields

$$\beta_i = (\mathbf{X}^T \mathbf{X})^{-1} \left[\sum_{j=1}^n \mathbf{x}_j y_{ij} - \frac{1}{\sum_{i=1}^m \frac{1}{\sigma_i^2}} \sum_{i=1}^m \frac{1}{\sigma_i^2} \sum_{j=1}^n \mathbf{x}_j y_{ij} \right] + \bar{\beta}^{(w)}, \quad (103)$$

where $\bar{\beta}^{(w)}$ is defined in (66).

Taking partial derivatives of σ_i^2 and setting the result to zero gives

$$\sigma_i^2 = \frac{1}{n} \sum_{j=1}^n (y_{ij} - \beta_{0i} - \mathbf{x}_j^T \beta_i - d_j)^2. \quad (104)$$

Substituting (101) into (104) yields

$$\sigma_i^2 = \frac{1}{n} \sum_{j=1}^n \left(y_{ij} - \bar{y}_{i\cdot} - \bar{y}_{\cdot j}^{(w)} + \bar{y}^{(w)} - \mathbf{x}_j^T \beta_i + \mathbf{x}_j^T \bar{\beta}^{(w)} \right)^2, \quad (105)$$

where $\bar{y}_{i\cdot}$, $\bar{y}_{\cdot j}^{(w)}$ and $\bar{y}^{(w)}$ are defined in (15), (11) and (18), respectively.

Given initial estimates for $\bar{\beta}^{(w)}$ and $\{\sigma_i^2\}_{i=1}^m$, estimates for β_i and σ_i^2 can then be iteratively updated using equations (103), (105), and (66) until convergence.

After estimating σ_i^2 's, they can then be shrunk (squeezed) toward the common noise variance to obtain robust estimates for σ_i^2 , as done in Section 3.4.

5 EXPERIMENTS

We evaluate the performance of the proposed algorithms. To save space, we only verify the proposed algorithm for the simple regression model (2) (referred to as slr-ADMM). We set the penalty parameter as $\alpha = 0.01\alpha_{\max}$, where α_{\max} is determined according to (47).

5.1 Simple Test with Synthetic Data

We simulate RNA-seq data with a total of $m = 1000$ genes and $n = 15$ samples. Other parameters are set as below.

Table 1
Nonlinear Model Results

parameter	meaning
$\beta_{0i} \sim \mathcal{N}(-3, 2)$	y -intercept
$\beta_i = 0$	log-fold change for non-DE genes
$\beta_i \sim \mathcal{N}(2, 1)$	log-fold change for up-regulated DE genes
$\beta_i \sim \mathcal{N}(-2, 1)$	log-fold change for down-regulated DE genes
$\mathbf{x}_j \sim \mathcal{N}(0, 1)$	condition data
$y_{ij} \sim \mathcal{N}(\beta_{0i} + \beta_i \mathbf{x}_j, 0.1)$	log gene expression
$\log(l_i) \sim \text{unif}(5, 10)$	log gene lengths
$N_j \sim \text{Unif}(3, 5) \times 10^7$	library sizes
$c_{ij} = \left\lfloor N_j \cdot \frac{l_i e^{y_{ij}}}{\sum_{i=1}^m l_i e^{y_{ij}}} \right\rfloor + 1$	read counts

We first simulate with 700 non-DE genes and 300 DE genes. Among DE genes 50% are up-regulated while the remaining 50% are down-regulated. The fitted $\{\beta_i\}_{i=1}^m$ using slr-ADMM are plotted in Figure 1(a). We see that the non-DE genes corresponds to exactly zero or close-to-zero coefficients while DE genes corresponds to large non-zero coefficients and they are easily distinguishable from each other. In Figure 1(b), we increase the percent of up-regulated DE genes to 70%. Our method still retrieves almost all non-zero β_i 's while shrinking all other β_i 's to zero. We further increase the percent of up-regulated DE genes to 90%, for which our method still achieves accurate estimates [Figure 1(c)].

In Figure 1(d-f), we increase the number of DE genes to 500, among which 50%, 70% or 90% are up-regulated while others are down-regulated. Our method still achieves accurate estimates. In Figure 1(g-h), we further increase the number of DE genes to 700 among which 50% or 70% are up-regulated, for which our method still achieves accurate estimates when. Only when we simulate with 700 DE genes among which 90% are up-regulated, our method fails to distinguish between DE and non-DE genes since the estimated regression coefficients of the latter are not zero either [Figure 1(i)].

Using a different gene expression measure such as CPM, RPKM or TPM values computed with formulas in (1) yields essentially the same result.

5.2 Comparison with Existing Methods with Real Data

We compare our method with edgeR-robust [20], [29], DESeq2 [18], and voom [16], [17], all of which are state-of-the-art methods for detecting differential gene expression from RNA-seq data.

We use RNA-seq data with a total of $m = 1000$ genes and $n = 15$ samples. We simulate both log-normally distributed read counts, which is the model assumptions of

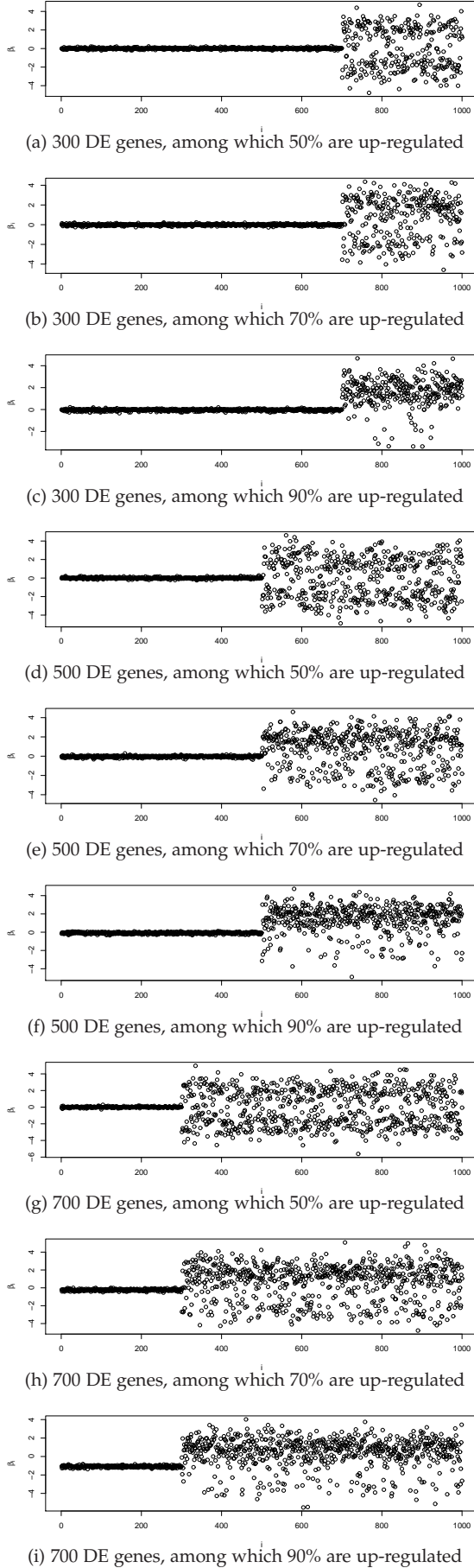


Figure 1. Estimated β_i from simulated data with simple linear regression model. The number of genes is $m = 1000$ and number of samples is $n = 15$. The penalty parameter is set as $\alpha = 0.01\alpha_{\max}$, where α_{\max} is determined according to (47).

voom and slr-ADMM, as well as negative-binomially distributed read counts, which is the underlying assumption of edgeR-robust and DESeq2. The gene expression levels and library sizes for both simulations are generated based on a real RNA-seq dataset [30]. The read counts are simulated according to [20]. We slightly modify the simulator to allow log-normally distributed data, as well as variable fold changes. We simulate data sets with 30%, 50%, 70%, or 90% DE genes, among which 50%, 70% or 90% are up-regulated while others are down-regulated. The log-fold changes for unit changes of explanatory variables, i.e., β_i 's, for up-regulated DE genes are assumed to be distributed as $\mathcal{N}(1, 1)$, while that for down-regulated DE genes are distributed as $\mathcal{N}(-1, 1)$. We consider high signal-to-noise ratio (low noise level) scenarios³, and set the negative-binomial dispersion parameter to one fifth of that generated from the real RNA-seq dataset. For log-normal data simulation, log read counts are assumed to be normally distributed with $\sigma = 0.1$.

The AUCs for DE gene detection from log-normally and negative-binomially distributed data using all four methods are summarized in Tables 2 and 3, respectively.

In Table 2, we see that the voom and slr-ADMM perform the best with log-normally distributed data. In relatively easier cases where only a small proportion of genes are differentially expressed, in a symmetric manner (which means the numbers of up- and down-regulated genes among DE genes are roughly the same), the voom and DESeq2 perform the best. The slr-ADMM is slightly inferior to the best performer but by only a small margin (i.e., within approximately one standard error for the log-normal distributed datasets). In challenging cases where a large proportion of genes are differentially expressed in a asymmetric manner (e.g., when more than 75% genes are up-regulated), the slr-ADMM performs the best. And with the increase of the percent of DE genes and/or percent of up-regulated genes, the performance gain of the slr-ADMM over completing methods increases and becomes more significant.

For negative-binomially distributed data, in Table 3 we see that the edgeR-robust and DESeq2 perform the best in easy cases. But again, in challenging scenarios where a large proportion of genes are differentially expressed in a asymmetric manner, the slr-ADMM is superior to all other methods.

Note that when more samples are available (e.g., $n = 25$), the performance gain of the slr-ADMM over completing methods becomes even more significant, for both log-normally and negative-binomially distributed data. For sake of conciseness, the results are not shown here.

6 DISCUSSION

A unified statistical model is proposed for joint between-sample normalization and DE detection of RNA-seq data. The sample-specific normalization factors are modeled as unknown parameters and jointly estimated together with DE detection. As a result, the model is robust against normalization errors and is independent of the units (i.e.,

3. For low SNRs, the simulation results are essentially similar except that the relative performance gain or loss becomes less significant and non-typical, which renders it difficult to interpret.

Table 2

AUC comparison of edgeR-robust, DESeq2, limma and slr-ADMM in log-normally distributed data. The number of samples is $n = 15$. The table shows the percent of DE genes (DE %), percent of up-regulated genes among all the DE genes (Up %), as well as the mean AUCs for all four methods measured using 10 simulated replicates. The standard errors of the mean AUCs are given in parentheses.

DE (%)	Up (%)	edgeR-robust	DESeq2	voom	slr-ADMM
30	50	0.954 (0.0023)	0.955 (0.0021)	0.962 (0.002)	0.9604 (0.0016)
30	70	0.9476 (0.004)	0.9506 (0.0039)	0.9585 (0.004)	0.9615 (0.004)
30	90	0.9298 (0.0054)	0.9365 (0.0029)	0.9404 (0.0043)	0.9554 (0.0018)
50	50	0.9498 (0.0023)	0.9516 (0.0022)	0.9607 (0.0023)	0.9593 (0.0022)
50	70	0.9214 (0.0031)	0.9333 (0.0024)	0.9357 (0.0029)	0.9558 (0.0029)
50	90	0.8661 (0.0066)	0.8852 (0.0045)	0.8926 (0.0055)	0.9499 (0.0013)
70	50	0.9482 (0.0023)	0.9498 (0.0023)	0.9574 (0.0019)	0.9564 (0.0023)
70	70	0.8556 (0.0031)	0.8925 (0.0036)	0.8818 (0.0033)	0.9467 (0.0023)
70	90	0.6936 (0.0083)	0.7598 (0.0074)	0.7223 (0.0074)	0.8587 (0.0074)

Table 3

AUC comparison of edgeR-robust, DESeq2, limma and slr-ADMM in negative-binomially distributed data. See Table 2 for descriptions.

DE (%)	Up (%)	edgeR-robust	DESeq2	voom	slr-ADMM
30	50	0.8935 (0.0051)	0.8942 (0.0046)	0.8941 (0.0042)	0.8909 (0.0042)
30	70	0.892 (0.0038)	0.8914 (0.0046)	0.8897 (0.0043)	0.8915 (0.0045)
30	90	0.8711 (0.0047)	0.8707 (0.0034)	0.868 (0.0046)	0.8875 (0.0038)
50	50	0.9116 (0.0024)	0.9119 (0.0027)	0.9109 (0.0024)	0.9074 (0.0025)
50	70	0.8741 (0.0043)	0.8751 (0.0032)	0.8702 (0.0043)	0.8872 (0.0034)
50	90	0.8096 (0.0052)	0.8166 (0.0035)	0.8043 (0.006)	0.8695 (0.0034)
70	50	0.9039 (0.0023)	0.9024 (0.0019)	0.9024 (0.002)	0.8992 (0.0018)
70	70	0.8493 (0.0044)	0.8423 (0.0054)	0.8421 (0.0047)	0.8742 (0.0028)
70	90	0.661 (0.0079)	0.661 (0.0073)	0.6535 (0.0073)	0.7375 (0.0067)

counts, CPM/RPM, RPKM/FPKM or TPM) in which gene expression levels are summarized.

For the model with a single treatment condition, we introduce the ℓ_1 -norm penalty to the linear regression model. The ℓ_1 -norm penalty favors sparse solutions (forces some coefficients to be exactly zero). This is desirable since many genes are not differentially expressed. From a Bayesian point of view, the lasso penalty corresponds to a Laplace (double exponential, with zero-mean) prior over the regression coefficients. By contrast, existing methods do not exploit the sparsity-inducing prior information. For the model with multiple treatment conditions, two types of penalty functions are introduced. In the first one only one covariate is of interest while all other covariates are treated as confounding factors. We are interested in testing whether that specific covariate is associated with differential expression. In the second case all covariates are of interest (there are

no confounding covariates) and we are interested in testing whether any covariate affects the differential expression of a gene. Regarding choice of the penalty parameter, we theoretically derive the maximum penalty parameter α_{\max} that leads to all-zero solution, and set $\alpha = \epsilon \alpha_{\max}$ with $0 < \epsilon < 1$. Empirically we found that the performance is not sensitive to ϵ and setting $\epsilon = 0.01$ works quite well in a wide range of parameter settings. This avoids computationally expensive cross validation procedure to tune the penalty parameter.

Simulation studies show that the proposed methods perform better than or comparably to existing methods in terms of AUC. The performance gain is more significant when a large proportion of genes (e.g., more than half) are differentially expressed and/or the up- and down-regulated DE genes are unbalanced in number, particularly in the presence of high signal-to-noise ratios or large sample-size.

The R as well as MATLAB codes of the algorithms described in the paper are available for download at <http://www-personal.umich.edu/~jianghui/lr-ADMM/>.

The authors would like to thank...

REFERENCES

- [1] A. Mortazavi, B. A. Williams, K. McCue, L. Schaeffer, and B. Wold, "Mapping and quantifying mammalian transcriptomes by RNA-Seq," *Nat Methods*, vol. 5, no. 7, pp. 621–628, Jul 2008.
- [2] C. Trapnell, B. A. Williams, G. Pertea, A. Mortazavi, G. Kwan, M. J. van Baren, S. L. Salzberg, B. J. Wold, and L. Pachter, "Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation," *Nat Biotechnol*, vol. 28, no. 5, pp. 511–515, May 2010.
- [3] H. Jiang and W. H. Wong, "Statistical inferences for isoform expression in RNA-Seq," *Bioinformatics*, vol. 25, no. 8, pp. 1026–1032, Apr 2009.
- [4] J. Salzman, H. Jiang, and W. H. Wong, "Statistical modeling of RNA-Seq data," *Statistical Science*, vol. 26, no. 1, pp. 62–83, 2011.
- [5] Z. Wang, M. Gerstein, and M. Snyder, "RNA-Seq: a revolutionary tool for transcriptomics," *Nat Rev Genet*, vol. 10, no. 1, pp. 57–63, Jan 2009.
- [6] M.-A. Dillies, A. Rau, J. Aubert, C. Hennequet-Antier, M. Jeanmougin, N. Servant, C. Keime, G. Marot, D. Castel, J. Estelle, G. Guernec, B. Jagla, L. Jouneau, D. Laloe, C. Le Gall, B. Schaeffer, L. ffer, S. Le Crom, M. Guedj, F. Jaffrezic, and F. S. C. , "A comprehensive evaluation of normalization methods for illumina high-throughput RNA sequencing data analysis," *Brief Bioinform*, vol. 14, no. 6, pp. 671–683, Nov 2013.
- [7] F. Rapaport, R. Khanin, Y. Liang, M. Pirun, A. Krek, P. Zumbo, C. E. Mason, N. D. Socci, and D. Betel, "Comprehensive evaluation of differential gene expression analysis methods for RNA-seq data," *Genome Biology*, vol. 14, no. 9, p. R95, Sep 2013.
- [8] M. D. Robinson and A. Oshlack, "A scaling normalization method for differential expression analysis of RNA-seq data," *Genome Biol*, vol. 11, no. 3, p. R25, 2010.
- [9] B. M. Bolstad, R. A. Irizarry, M. Astrand, and T. P. Speed, "A comparison of normalization methods for high density oligonucleotide array data based on variance and bias," *Bioinformatics*, vol. 19, no. 2, pp. 185–193, Jan 2003.
- [10] G. K. Smyth, "Limma: linear models for microarray data," in *Bioinformatics and computational biology solutions using R and Bioconductor*. Springer, 2005, pp. 397–420.
- [11] J. H. Bullard, E. Purdom, K. D. Hansen, and S. Dudoit, "Evaluation of statistical methods for normalization and differential expression in mRNA-Seq experiments," *BMC Bioinformatics*, vol. 11, p. 94, 2010.
- [12] S. Anders and W. Huber, "Differential expression analysis for sequence count data," *Genome Biol*, vol. 11, no. 10, p. R106, 2010.
- [13] B. Li, V. Ruotti, R. M. Stewart, J. A. Thomson, and C. N. Dewey, "RNA-Seq gene expression estimation with read mapping uncertainty," *Bioinformatics*, vol. 26, no. 4, pp. 493–500, Feb 2010.

- [14] A. Oshlack, M. J. Wakefield *et al.*, "Transcript length bias in RNA-seq data confounds systems biology," *Biol Direct*, vol. 4, no. 1, p. 14, 2009.
- [15] J. C. Marioni, C. E. Mason, S. M. Mane, M. Stephens, and Y. Gilad, "RNA-seq: an assessment of technical reproducibility and comparison with gene expression arrays," *Genome research*, vol. 18, no. 9, pp. 1509–1517, 2008.
- [16] C. W. Law, Y. Chen, W. Shi, and G. K. Smyth, "Voom: precision weights unlock linear model analysis tools for RNA-seq read counts," *Genome Biol*, vol. 15, no. 2, p. R29, 2014.
- [17] M. E. Ritchie, B. Phipson, D. Wu, Y. Hu, C. W. Law, W. Shi, and G. K. Smyth, "limma powers differential expression analyses for RNA-sequencing and microarray studies," *Nucleic Acids Research*, vol. 43, no. 7, p. e47, Jan 2015.
- [18] M. I. Love, W. Huber, and S. Anders, "Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2," *Genome Biology*, vol. 15, no. 12, p. 550, 2014.
- [19] H. Jiang and J. Salzman, "A penalized likelihood approach for robust estimation of isoform expression," *Statistics and Its Interface*, vol. 8, no. 4, pp. 437–445, 2015.
- [20] X. Zhou, H. Lindsay, and M. D. Robinson, "Robustly detecting differential expression in RNA sequencing data using observation weights," *Nucleic acids research*, vol. 42, no. 11, pp. e91–e91, 2014.
- [21] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [22] H. Jiang and T. Zhan, "Unit-free and robust detection of differential expression from RNA-Seq data," *Submitted for publication*, 2016.
- [23] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.
- [24] H. Ji and W. H. Wong, "TileMap: create chromosomal map of tiling array hybridizations," *Bioinformatics*, vol. 21, no. 18, pp. 3629–3636, 2005.
- [25] H. Ji and X. S. Liu, "Analyzing 'omics data using hierarchical models," *Nature biotechnology*, vol. 28, no. 4, pp. 337–340, 2010.
- [26] G. Smyth, "Statistical applications in genetics and molecular biology," *Linear models and empirical Bayes methods for assessing differential expression in microarray experiments*, 2004.
- [27] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of statistical software*, vol. 33, no. 1, p. 1, 2010.
- [28] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.
- [29] M. D. Robinson, D. J. McCarthy, and G. K. Smyth, "edgeR: a bioconductor package for differential expression analysis of digital gene expression data," *Bioinformatics*, vol. 26, pp. 139–140, Jan 2010.
- [30] J. K. Pickrell, J. C. Marioni, A. A. Pai, J. F. Degner, B. E. Engelhardt, E. Nkadori, J.-B. Veyrieras, M. Stephens, Y. Gilad, and J. K. Pritchard, "Understanding mechanisms underlying human gene expression variation with RNA sequencing," *Nature*, vol. 464, no. 7289, pp. 768–772, 2010.

$$L_\rho(\boldsymbol{\beta}, \delta_0, \lambda) = \sum_{i=1}^m \frac{1}{2\sigma_i^2} \sum_{j=1}^n (\tilde{y}_{ij} - x_j \beta_i + x_j \delta_0)^2 + \alpha \sum_{i=1}^m |\beta_i| + \lambda \left(\frac{1}{\sum_{i=1}^m \frac{1}{\sigma_i^2}} \sum_{i=1}^m \frac{1}{\sigma_i^2} \beta_i - \delta_0 \right) + \frac{\rho}{2} \left(\frac{1}{\sum_{i=1}^m \frac{1}{\sigma_i^2}} \sum_{i=1}^m \frac{1}{\sigma_i^2} \beta_i - \delta_0 \right)^2. \quad (28)$$

$$\beta_i = \frac{\sigma_i^2 \left(\sum_{\ell=1}^m \sigma_\ell^{-2} \right)^2}{\sigma_i^2 \left(\sum_{\ell=1}^m \sigma_\ell^{-2} \right)^2 + \rho} T_{\sigma_i^2} \alpha \left[\left(\sum_{j=1}^n x_j \tilde{y}_{ij} + \delta_0 \right) - \frac{\rho}{\sum_{\ell=1}^m \frac{1}{\sigma_\ell^2}} \left(\frac{1}{\sum_{\ell=1}^m \frac{1}{\sigma_\ell^2}} \sum_{\ell \neq i} \frac{1}{\sigma_\ell^2} \beta_\ell - \delta_0 + \frac{\lambda}{\rho} \right) \right], \quad (30)$$

$$L_\rho(\{\boldsymbol{\beta}_i\}, \delta_0, \boldsymbol{\lambda}) = \sum_{i=1}^m \frac{1}{2\sigma_i^2} \sum_{j=1}^n \left(\tilde{y}_{ij} - \mathbf{x}_j^T \boldsymbol{\beta}_i + \mathbf{x}_j^T \boldsymbol{\delta}_0 \right)^2 + \alpha \sum_{i=1}^m |\beta_{ip}| + \boldsymbol{\lambda}^T \left(\frac{1}{\sum_{i=1}^m \frac{1}{\sigma_i^2}} \sum_{i=1}^m \frac{1}{\sigma_i^2} \boldsymbol{\beta}_i - \boldsymbol{\delta}_0 \right) + \frac{\rho}{2} \left\| \frac{1}{\sum_{i=1}^m \frac{1}{\sigma_i^2}} \sum_{i=1}^m \frac{1}{\sigma_i^2} \boldsymbol{\beta}_i - \boldsymbol{\delta}_0 \right\|^2. \quad (65)$$

$$L_\rho(\{\boldsymbol{\beta}_i\}, \delta_0, \boldsymbol{\lambda}) = \sum_{i=1}^m \frac{1}{2\sigma_i^2} \sum_{j=1}^n \left(\tilde{y}_{ij} - \mathbf{x}_j^T \boldsymbol{\beta}_i + \mathbf{x}_j^T \boldsymbol{\delta}_0 \right)^2 + \alpha \sum_{i=1}^m \|\boldsymbol{\beta}_i\| + \boldsymbol{\lambda}^T \left(\frac{1}{\sum_{i=1}^m \frac{1}{\sigma_i^2}} \sum_{i=1}^m \frac{1}{\sigma_i^2} \boldsymbol{\beta}_i - \boldsymbol{\delta}_0 \right) + \frac{\rho}{2} \left\| \frac{1}{\sum_{i=1}^m \frac{1}{\sigma_i^2}} \sum_{i=1}^m \frac{1}{\sigma_i^2} \boldsymbol{\beta}_i - \boldsymbol{\delta}_0 \right\|^2. \quad (76)$$

$$\begin{aligned} L_i(\{\boldsymbol{\beta}_i\}, \delta_0, \boldsymbol{\lambda}) &= \frac{1}{2\sigma_i^2} \sum_{j=1}^n \left(\tilde{y}_{ij} - \mathbf{x}_j^T \boldsymbol{\beta}_i + \mathbf{x}_j^T \boldsymbol{\delta}_0 \right)^2 + \alpha \|\boldsymbol{\beta}_i\| + \boldsymbol{\lambda}^T \frac{1}{\sum_{\ell=1}^m \frac{1}{\sigma_\ell^2}} \frac{1}{\sigma_i^2} \boldsymbol{\beta}_i + \frac{\rho}{2} \left(\frac{1}{\sum_{\ell=1}^m \frac{1}{\sigma_\ell^2}} \frac{1}{\sigma_i^2} \boldsymbol{\beta}_i + \frac{1}{\sum_{\ell=1}^m \frac{1}{\sigma_\ell^2}} \sum_{\ell \neq i} \frac{1}{\sigma_\ell^2} \boldsymbol{\beta}_\ell - \boldsymbol{\delta}_0 \right)^2 \\ &= \frac{1}{2} \boldsymbol{\beta}_i^T \left(\frac{1}{\sigma_i^2} \mathbf{X}^T \mathbf{X} + \frac{\rho}{\sigma_i^4 \left(\sum_{\ell=1}^m \frac{1}{\sigma_\ell^2} \right)^2} \mathbf{I}_p \right) \boldsymbol{\beta}_i - \boldsymbol{\beta}_i^T \mathbf{v}_i + \alpha \|\boldsymbol{\beta}_i\| + c, \end{aligned} \quad (77)$$
